Др Ђорђе Дугошија

СИМПЛЕКС МЕТОДА

У претходним радовима [2] и [3] описана је теорија линеарног програмирања. У овом раду приказаћемо једну од основних метода за решавање проблема линеарног програмирања – *симплекс методу*. Иако је од њеног открића прошло више од пола века (Dancig [1]), у нашој средини се још увек на ту методу гледа као на веома специјално и сложено математичко знање. У настави се налази искључиво на факултетима математичког и економског профила, често нејасно, компликовано или некоректно изложена. Циљ овога рада је да се на минималном броју страна прикажу и „педагошке“ и „компјутерске“ варијанте симплекс методе и да тако она постане доступна свима који имају основно универзитетско математичко образовање.

Проблем који се решава симплекс методом је налажење минимума (или максимума) линеарне функције

$$f(x) = cx$$

(c је позната врста, $x = (x_1, x_2, \ldots, x_n)^\top$ је колона са n непознатих) при ограничењима

(1) $\quad Ax = b \quad$ (A је позната $m \times n$ матрица, b је позната колона)

и

(2) $$x \geqslant 0.$$

Другим речима, тражи се минимум линеарне функције од n непознатих на скупу ненегативних решења система (1) линеарних једначина.

Облик (1)-(2) проблема линеарног програмирања зове се *канонски*.

Сваки проблем линеарног програмирања може се свести на канонски облик, при чему се може претпоставити да је $b \geqslant 0$. Заиста, ако проблем има неједначину облика $ax \leqslant \beta$, она се може заменити са $ax + y = \beta$, $y \geqslant 0$, а неједначина облика $ax \geqslant \beta$ са $ax - y = \beta$, $y \geqslant 0$, при чему је y нова ненегативна променљива, која се зове *изравнавајућа* за посматрано ограничење. Ако у проблему недостаје услов ненегативности неке променљиве, она се може заменити разликом две нове ненегативне променљиве, јер је сваки број разлика два ненегативна броја.

Главна предност канонског облика над осталим облицима је у униформном и упрошћеном запису ограничења типа неједнакости, као и у могућношћу решавања система (1) линеарних једначина, што се у симплекс методи ефикасно користи.

Систем (1) може се посматрати као проблем да се колона b изрази као линеарна комбинација колона матрице A са тежинама x_j, $j = 1, 2, \dots, n$:

$$b = \sum_{j=1}^{n} x_j k_j \tag{1}$$

Из линеарне алгебре је познато да све линеарне комбинације колона матрице граде векторски простор. Максимални подскупови линеарно независних колона матрице су базе тог простора. Све базе имају исти број елемената једнак рангу r матрице A.

Систем (1) има решења ако b припада простору колона. Тада се b као и небазисне колоне матрице могу на јединствен начин изразити као линеарне комбинације колона изабране базе (базисних колона). Сва решења система (1) добијају се кад се слободно изаберу непознате уз небазисне колоне и преостали систем реши по непознатим уз базисне колоне, што је на јединствен начин могуће.

Подматришу матрице састављену од колона неке базе простора колона зваћемо *база матрице*, а променљиве које одговарају тим колонама зваћемо *базисне променљиве*. Колону састављену од базисних променљивих обележаваћемо са x_B. Са N ћемо обележити подматрицу састављену од небазисних колона, а са x_N колону од небазисних променљивих. Систем (1) може се тада написати у облику

$$Bx_B + Nx_N = b \tag{1}$$

и, ако има решења, сва његова решења имају компоненте x_B и x_N при чему се x_N може слободно изабрати док се базисне променљиве за било који избор x_N могу израчунати на следећи начин.

Како B не мора бити квадратна матрица, помножимо обе стране у (1) најпре са $B^\top$, а затим са $(B^\top B)^{-1}$ (што постоји, јер су колоне у B линеарно независне). Добијамо:

$$x_B = (B^\top B)^{-1}[B^\top b - B^\top N x_N].$$

Решење које одговара избору $x_N = 0$ зове се *базисно решење* за базу B. Базу за коју је припадно базисно решење ненегативно (дакле допустива тачка проблема) зваћемо *допустива база*.

И функцију циља можемо изразити искључиво преко небазисних променљивих заменом израчунатог x_B у израз за f:

$$f = c_B x_b + c_N x_N = c_B(B^\top B)^{-1}[B^\top b - B^\top N x_N] + c_N x_N = f_0 + r_N x_N, \tag{3}$$

где је $f_0 = c_B(B^\top B)^{-1}B^\top b$ и $r_N = c_N - c_B(B^\top B)^{-1}B^\top N$.

У случају да је матрица B инвертибилна, претходне формуле се упрошћавају, јер је $(B^\top B)^{-1}B^\top = B^{-1}$.

Из (3) следи једноставан

Довољан услов оптималности. *Ако је базисно решење $\overline{x}$ за базу B допустиво, а сви коефицијенти r_N уз небазисне променљиве ненегативни, оно је оптимално решење полазног проблема.*

Заиста, из (3) следи $f(x) \geqslant f_0 = f(\overline{x})$ за сва допустива решења x проблема.

Ако $\overline{x}$ није прошло тест оптималности, r_N има бар једну негативну координату r_j.

Поправљање решења

Можемо потражити боље решење као функцију ненегативног параметра $x_j = t$, задржавајући остале слободне променљиве из x_N на вредности нула. За такав избор слободних променљивих имамо

$$x_B(t) = \overline{x}_B - (B^\top B)^{-1} B^\top k_j t \tag{4}$$

и

$$f(t) = f_0 + r_j t. \tag{5}$$

Из последње везе види се да са растењем параметра t долази до опадања вредности функције циља. Највише ћемо снизити функцију циља ако бирамо највеће ненегативно t за кога добијемо допустиве тачке тј. за кога је

$$x_B(t) \geqslant 0. \tag{6}$$

Приметимо да је (6) систем линеарних неједначина са једном непознатом и да се може написати у облику

$$\overline{x}_B \geqslant yt, \tag{6'}$$

при чему је y стубац коефицијената којима се изражава колона k_j у бази B, тј. такав да је $k_j = By$ (што, решено по y, даје $y = (B^\top B)^{-1} B^\top k_j$).

Ако је $y \leqslant 0$, (6) је задовољено за све $t \geqslant 0$. Решење $x(t)$ са координатама $x_B = \overline{x}_B - (B^\top B)^{-1} B^\top k_j t$, $x_j = t$ и осталим координатама једнаким нули је допустиво за све $t \geqslant 0$ и $f(t) \to -\infty$ кад $t \to \infty$. Тиме је проблем линеарног програмирања у овом случају решен.

Ако y има бар једну позитивну координату, систем (6) има решење

$$0 \leqslant t \leqslant \hat{t},$$

где је $\hat{t} = \min\Big\{ \dfrac{\overline{x}_s}{y_s} \mid y_s > 0,\ k_s \in B \Big\}$. Нека је $x_i(\hat{t}) = \overline{x}_i - \hat{t} y_i = 0$ било која координата вектора $x_B(\hat{t})$ која се анулира и за коју важи $y_i > 0$. Скуп колона $B^1 = (B \setminus \{k_i\}) \cup \{k_j\}$ поново чини базу простора колона, јер се k_j у бази B изражава са коефицијентом y_i уз k_i различитим од нуле (позитивним). Базисно решење које одговара бази B^1 је управо $\hat{x} = x(\hat{t})$, које због $f(\hat{x}) = f(\overline{x}) + r_j \hat{t} \leqslant f(\overline{x})$ није лошије од $\overline{x}$.

Алгоритам симплекс методе

Корак 0. Полазимо од неке базе чије је припадно базисно решење допустиво.

Корак 1. Проверимо оптималност тог решења помоћу наведеног довољног услова. Ако решење задовољава услов оптималности, проблем је решен. У случају да решење не задовољава довољан услов, идемо на

Корак 2. Тражимо боље решење на описани начин.

Ако нађемо зрак дуж кога функција циља неограничено опада, проблем је решен. У супротном налазимо другу допустиву базу која се од прве разликује у једној колони и чије припадно базисно решење по вредности функције циља није лошије од претходног и идемо поново на корак 1.

Водећи рачуна да се базе не понове, поступак ће бити коначан, јер је број база највише $\binom{n}{r}$. Понављање база (циклирање) се може избећи посебним процедурама бирања улажења и излажења колона из база. Једна од најпростијих таквих процедура је

Блендово правило минималног индекса ([4]):

У свакој итерацији за излаз из базе и за улаз у базу изабрати кандидата са најмањим индексом.

Доказ је сложен па га изостављамо (види на пример [6] или [7]).

Без примене процедуре против циклирања, понављање база је могуће, али је у пракси врло ретко. Доказано је да је вероватноћа појаве циклирања при случајно изабраним коефицијентима проблема једнака нули, те већина рачунарских имлементација симплекс методе не води о том рачуна.

Пример 1. Решити симплекс методом проблем

$$\min f = x - y$$

п.о.

$$x + y \leqslant 3, \quad 2x + y \geqslant 4, \quad y \geqslant 0.$$

Проблем прво трансформишемо у канонски облик уводећи изравнавајуће променљиве a и b и замењујући x са разликом две ненегативне променљиве x' и x'':

$$\min f = x' - x'' - y$$

п.о.

$$x' - x'' + y + a = 3, \quad 2x' - 2x'' + y - b = 4, \quad x', x'', y, a, b \geqslant 0.$$

Матрица система је

$$A = \begin{pmatrix} 1 & -1 & 1 & 1 & 0 \\ 2 & -2 & 1 & 0 & -1 \end{pmatrix}.$$

Једну допустиву базу простора колона чине прва и последња колона. Заиста за $B = \begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix}$ небазисна подматрица је $N = \begin{pmatrix} -1 & 1 & 1 \\ -2 & 1 & 0 \end{pmatrix}$, а припадно базисно решење, добијено избором $x'' = y = a = 0$ и решавањем система по x' и b, је

$(x', x'', y, a, b)^\top = (3,0,0,0,2)^\top \geqslant (0\ 0\ 0\ 0\ 0)^\top$. Како је

$$B^{-1} = B, \quad B^{-1}N = \begin{pmatrix} -1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix},$$

$$r_N = c_N - c_B B^{-1} N = (-1\ \ -1\ 0) - (1\ 0) \begin{pmatrix} -1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} = (0\ \ -2\ \ -1),$$

кандидати за улаз у базу су трећа (уз y) и четврта (уз a) колона матрице A. Сагласно Блендовом правилу, изабраћемо трећу колону за улаз у базу. Одредимо репрезентацију треће колоне у бази B решавањем система

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = By.$$

Добијамо $y = B^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Одредимо сада највеће t за које је $\overline{x}_B - yt = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - t \begin{pmatrix} 1 \\ 1 \end{pmatrix} \geqslant \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. То је $\hat{t} = \min\{\frac{3}{1}, \frac{2}{1}\} = 2$. При избору $t = \hat{t}$ анулира се друга од ових неједначина (која одговара променљивој b), те је шеста колона (уз b) (једини) кандидат за излаз из базе. Нова допустива база је зато $B = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$, а припадно базисно решење је $\hat{x} = (3 - \hat{t}, 0, \hat{t}, 0, 0)^\top = (1,0,2,0,0)^\top$. Даље је $B^{-1} = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}$, $r_N = (-1\ 0\ 0) - (1\ \ -1) \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 \\ -2 & 0 & -1 \end{pmatrix} = (0\ 3\ 2) \geqslant (0\ 0\ 0)$, па је $\hat{x}$, према довољном услову оптималности, оптимално решење проблема.

Варијанте симплекс методе

Изложени алгоритам симплекс методе је „сиров“ јер у свакој итерацији методе треба вршити инвертовање матрица, што није лак посао ни ручно ни рачунарски. Инвертовање је тривијално ако је база јединична матрица. Осим тога репрезентација небазисних колона преко јединичних базисних је непосредно читљива, а изражавање функције циља преко небазисних променљивих врло лако. У ту сврху довољно је од функције циља одузети погодну линеарну комбинацију једначина система (1), бирајући за тежине бројеве супротне коефицијентима уз базисне променљиве. То даје идеју да се алгоритам измени тако да се у сваком кораку појављују јединичне матрице као базе.

Претпоставимо да је полазна допустива база јединична. Приметимо да су координате припадног базисног решења у том случају баш координате вектора b (евентуално пермутоване), те је дакле $b \geqslant 0$. Нека према алгоритму у базу треба да уђе колона $k_j = (a_{1j} a_{2j} \cdots a_{mj})^\top$ која има бар један позитиван елемент. По алгоритму, из базе треба да изађе јединична базисна колона са јединицом у i-тој врсти матрице A, при чему је i одређено са

$$\frac{b_i}{a_{ij}} = \min\left\{ \frac{b_s}{a_{sj}} \;\middle|\; a_{sj} > 0 \right\}. \tag{7}$$

Да би нова база постала јединична довољно је елементарним трансформацијама (које не мењају скуп допустивих решења) претворити колону k_j у јединичну са јединицом на пољу (i, j). Елемент a_{ij} зваћемо *пивот*. Да бисмо га претворили у јединицу поделимо обе стране i-те једначине са a_{ij}, што је коректно, јер је $a_{ij} \neq 0$. Да бисмо анулирали остале елементе j-те колоне, од s-те једначине одузмемо (по странама) умножак i-те једначине са a_{sj}. Слободан члан s-те једначине постаје $b_s - b_i \dfrac{a_{sj}}{a_{ij}}$ и, због (7), остаје ненегативан за све $s \neq i$. Зато је базисно решење које одговара новој јединичној бази ненегативно, па се алгоритам може наставити чишћењем функције циља од нове базисне променљиве x_j. Ово је могуће урадити одбијајући од израза за f умножак i-те једначине система са коефијентом уз x_j. Како се полазни проблем може записати матрично (таблично) и алгоритам представити низом трансформација матрица, изложена варијанта симплекс методе зове се *таблична симплекс метода*:

Таблична симплекс метода

Проблему

$$\min f = cx + f_0, \quad \text{п.о}, \quad Ax = b, \quad x \geqslant 0$$

придружимо таблицу

$$\begin{array}{ccc} & & x \\ f - f_0 & = & c \\ b & = & A \end{array}.$$

Из педагошких разлога (ради лаког дешифровања) задржали смо у таблици запис функције f, променљивих x и знакове $=$ иако се они обично не записују.

Таблица се зове *симплекс таблица*, ако у матрици A постоји базисна јединична подматрица чији је формат једнак броју врста у A, ако је $b \geqslant 0$ и ако су коефицијенти који одговарају базисним колонама у првој врсти (изразу за функцију циља) једнаки нули.

Таблична варијанта методе полази од једне симплекс таблице. Ако су елементи прве врсте са изузетком првог елемента ненегативни, припадно базисно решење је оптимално. У супротном изабере се колона уз један од тих негативних коефицијената за улаз у базу. Ако та колона нема позитивних елемената функција циља је неограничена одоздо. Ако колона има позитивних елемената, међу њима се одреди пивот као један од позитивних елемената за кога се достиже минимум у (7). Матрица се трансформише по следећем *мнемотехничком правилу*:

(i) врста пивота дели се пивотом;

(ii) остали елементи матрице мењају се тако, што се од старе вредности одузме прозвод пројекција тог елемента на врсту и колону пивота раздељен пивотом.

Тиме се поново добија симплекс таблица и алгоритам понавља. Ако се базе не понове, што се може осигурати Блендовим правилом, поступак је коначан.

Мана табличне симплекс метода је могућност нагомилавања грешака приликом итерација. Зато већина комерцијалних програма ради тзв. *ревидираном* *симплекс методом* која се од алгоритма који смо првобитно изложили разликује само у неколико финеса:

Ревидирана симплекс метода

Ревидирана симплекс метода полази од неке допустиве базе B. Да би се функција циља изразила преко небазисних променљивих, од израза за функцију циља одузима се линеарна комбинација једначина система са тежинама $u = (u_1\ u_2\ \cdots\ u_m)$ таквим да је

$$c_B = uB. \tag{8}$$

Овај систем је могуће решити, јер његова матрица система ($B^\top$) има линеарно независне врсте. Израз за функцију циља добија облик

$$f - ub = r_N x_N, \quad r_N = c_N - uN. \tag{9}$$

У случају да колона k_j треба да уђе у базу, да би се утврдило која колона треба да изађе из ње, треба одредити репрезентацију колоне k_j у старој бази B, тј. решити систем

$$k_j = By. \tag{10}$$

Даље алгоритам иде по већ описаним правилима (решавањем система (6') итд).

Уско грло ревидиране симплекс методе је решавање система (8) и (10) без инвертовања матрица. На срећу они се могу свести на решавање низа простијих система, ако се искористи факт да се нова база добија из старе заменом једне колоне j-те по реду неком другом колоном k. Зато се нова база добија као производ старе базе и матрице којој је j-та јединична колона замењена колоном k. Матрице које се од јединичне матрице разликују у једној колони зову се *ета матрице*. Ако је почетна база била јединична матрица, p-та база има облик производа ета матрица $E_1 E_2 \cdots E_p$, па се у p-тој итерацији алгоритма системи (8) и (10) могу решавати као низ система, сваки са једном ета матрицом:

$$c_B = (\cdots((uE_1)E_2)\cdots)E_p. \tag{8}$$

$$k_j = E_p(E_{p-1}(\cdots(E_1 y)\cdots)). \tag{10}$$

Како памћење ета матрице захтева много мање меморије него памћење пуне матрице, овом модификацијом добија се могућност обраде на рачунару проблема са десетинама хиљада променљивих и/или ограничења, што бројни комерцијални програми (на пример LINDO, BLP, GAMS и др.) користе.

Како стартовати?

Да би се стартовала симплекс метода у свим варијантама потребна је прва допустива база. У случају да она није одмах видљива, постоји више начина да се нађе. Алгоритми који то раде углавном користе саму симплекс методу примењену на неки помоћни проблем. Изложићемо велико М-методу Charnes-a ([5]). За друге методе погледати на пример [6] или [8].

Метода велико М

Ако у проблему (1), који је у канонском облику са $b \geqslant 0$, почетна допустива база није видљива, можемо посматрати придружен проблем

$$\min f = cx + (MM \cdots M)y$$

п.о.

$$Ax + y = b, \quad x, y \geqslant 0,$$

где је y колона вештачких ненегативних променљивих и M веома велики ненегативан параметар (већи од сваког коначног броја с којим се у току алгоритма упоређује).

Приметимо да матрица придруженог проблема има јединичну допустиву базу (базисне променљиве су y, а небазисне x), а припадно базисно допустиво решење је $x = 0$, $y = b$. Стога се придружени проблем може решавати симплекс методом.

Анализирајмо могуће завршетке симплекс методе уз претпоставку да нема циклирања:

а) Нека је симплекс метода завршила нашавши оптимално решење са компонентама $\hat{x}$ и $\hat{y}$. Тада:

(i) ако је $\hat{y} = 0$, онда је $\hat{x}$ оптимално решење полазног проблема. Заиста, према јакој теореми дуалности проблем дуалан придруженом

$$\max ub$$

п.о.

$$uA \leqslant c, \quad u \leqslant (MM \cdots M)$$

има оптимално решење $\hat{u}$ и $\hat{u}b = c\hat{x}$. Тачка $\hat{u}$ је допустива за дуал

$$\max ub$$

п.о.

$$uA \leqslant c$$

полазног проблема, а $\hat{x}$ је допустиво решење полазног проблема. Због $\hat{u}b = c\hat{x}$, $\hat{x}$ је оптимално решење полазног проблема.

(ii) ако је $\hat{y} \neq 0$, полазни проблем нема ниједно допустиво решење. У супротном, за довољно велико M, добијена минимална вредност функције циља придруженог проблема била би већа од вредности његове функције циља у допустивој тачки чије су компоненте x (допустива тачка за полазни проблем) и $y = 0$, што је контрадикција.

б) Ако је симплекс метода завршила нашавши допустиви зрак $(x_0, y_0) + t(\hat{x}, \hat{y})$, $t \geqslant 0$ дуж којег функција циља неограничено опада кад $t \to \infty$, тада је

$$Ax_0 + y_0 = b, \quad A\hat{x} + \hat{y} = 0$$

и, за довољно велико M, мора бити $\hat{y}=0$ и $c\hat{x}<0$, па је $x_0+t\hat{x}$ за $t\geqslant 0$ зрак допустивих тачака за полазни проблем дуж којег функција циља неограничено опада.

Пример 2. Решимо табличном симплекс методом проблем из примера 1.

Проблему одговара таблица

$$\begin{array}{ccccccc} & & x' & x'' & y & a & b \\ f-0 & = & 1 & -1 & -1 & 0 & 0 \\ 3 & = & 1 & -1 & 1 & 1 & 0 \\ 4 & = & 2 & -2 & 1 & 0 & -1 \end{array}$$

Како матрица система нема јединичну допустиву базу (има само једну јединичну колону уз изравнавајућу променљиву a), довољно је да уведемо једну ненегативну вештачку променљиву c уз коју у функцији циља стоји коефицијент M, а у матрици ограничења недостајућа јединична колона. Проширеном проблему тада одговара таблица

$$\begin{array}{cccccccc} & & x' & x'' & y & a & b & c \\ f-0 & = & 1 & -1 & -1 & 0 & 0 & M \\ 3 & = & 1 & -1 & 1 & 1 & 0 & 0 \\ 4 & = & 2 & -2 & 1 & 0 & -1 & 1 \end{array}$$

Очистимо функцију циља од базисне променљиве c, множећи последњу врсту са $-M$ и додајући је првој врсти матрице. Добијамо таблицу

$$\begin{array}{cccccccc} & & x' & x'' & y & a & b & c \\ f-4M & = & 1-2M & -1+2M & -1-M & 0 & M & 0 \\ 3 & = & 1 & -1 & 1 & 1 & 0 & 0 \\ 4 & = & 2 & -2 & 1 & 0 & -1 & 1 \end{array}$$

Како је параметар M јако велики, биће $1-2M<0$ и $-1-M<0$, па су кандидати за базисне променљиве x' и y. Према Блендовом правилу изабраћемо x'. Колона која одговара овој променљивој има два позитивна елемента међу којима тражимо пивот. Како је $\dfrac{3}{1}>\dfrac{4}{2}$, пивот је 2. Трансформацијом таблице по наведеном мнемотехничком правилу добијамо нову симплекс таблицу

$$\begin{array}{cccccccc} & & x' & x'' & y & a & b & c \\ f-2 & = & 0 & 0 & -3/2 & 0 & 1/2 & 1/2+M \\ 1 & = & 0 & 0 & 1/2 & 1 & 1/2 & -1/2 \\ 2 & = & 1 & -1 & 1/2 & 0 & -1/2 & 1/2 \end{array}$$

Коефицијент уз y у функцији циља је негативан, те пивот тражимо у колони испод y. Како је $\dfrac{1}{1/2}<\dfrac{2}{1/2}$, пивот је $1/2$ у трећој врсти. Трансформисана таблица је

$$\begin{array}{cccccccc} & & x' & x'' & y & a & b & c \\ 1 & = & 0 & 0 & 0 & 3 & 2 & M-1 \\ 2 & = & 0 & 0 & 1 & 2 & 1 & -1 \\ 1 & = & 1 & -1 & 0 & -1 & -1 & 1 \end{array}$$

Одавде читамо оптимално решење: $f_{\min}=-1$ за $(x',x'',y,a,b,c)=(1,0,2,0,0,0)$.

Коментар

Алгоритам симплекс методе има и популарну (али ретко где коретно формулисану) геометријску интерпретацију (види [6]).

Допустиви скуп је конвексан полиедарски скуп, а базисно допустиво решење (проблема у канонском облику) је његова екстремна тачка (врх), тј. тачка без које је остатак допустивог скупа и даље конвексан. Две екстремне тачке зову се *суседне*, ако уклањањем дужи које оне одређују, од допустивог скупа остаје конвексан скуп. У свакој итерацији симплекс метода полази из једног врха допустивог скупа и, ако пређе у другу допустиву тачку (а не мора), та тачка је њему суседни врх у коме функција циља није већа него у претходном. Циклирање одговара враћању у неки врх. Ако се примењује неко антициклинг правило, поступак се завршава у врху који је оптимално решење или се добија екстремна ивица (зрак) допустивог скупа дуж које функција циља неограничено опада.

По брзини рада симплекс метода спада у такозване *неполиномијалне алгоритме.* Конструисани су примери са n променљивих које симплекс метода решава у 2^n итерација. Зато се може очекивати да ће са порастом димензије проблема (броја непознатих и ограничења) доћи до проблема у раду. Пракса пак показује да се то ретко догађа. Разлог лежи у томе што су екстремно тешки случајеви ретки. Теоријски је показано да је симплекс алгоритам „у средњем“ полиномијалан. Ипак, трагање за ефикасним полиномијалним алгоритмом за решење проблема линеарног програмирања било је главни проблем у последње две декаде. Он је успешно окончан откривањем Кармакарове и сродних „унутрашњих“ метода (види [8]). И поред тога, симплекс метода остаје једна од главних метода за решавање проблема линеарног програмирања.

ЛИТЕРАТУРА

[1] Danzig, G., *Linear Programming and Extensions*, Princeton 1963.

[2] Дугошија, Ђ., *Фурије-Моцкинов метод елиминације*, Настава математике **XLV** 2000, св.1-2, 42–47.

[3] Дугошија, Ђ, *Теорија линеарног програмирања*, Настава математике **XLV** 2000, св.3-4, 42–48.

[4] Bland, R.G., *New finite pivoting rule for the simplex method*, Mathematics of Operations Researq **2**, 1977, 103–107.

[5] Charnes, A., *Optimality and degeneracy in linear programming*, Ekonometrica **20**, 1952, 160–170.

[6] Дугошија, Ђ., *Линеарно програмираље (скрипта)*, Математички факултет, Београд, 1994.

[7] Chvatal, V., *Linear Programming*, Freeman and Company, 1983.

[8] Цветковић, Д., Чангаловић, М., Дугошија, Ђ., Ковачевић-Вујчић, В., Симић, С., Вулета, Ј., *Комбинаторна оптимизација*, Допис, Београд 1996.